

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06110718 A**

(43) Date of publication of application: **22.04.94**

(51) Int. Cl.

**G06F 11/00**

**G06F 9/06**

**G06F 12/00**

(21) Application number: **04261336**

(71) Applicant: **TOSHIBA CORP**

(22) Date of filing: **30.09.92**

(72) Inventor: **SEKIDO KAZUNORI**

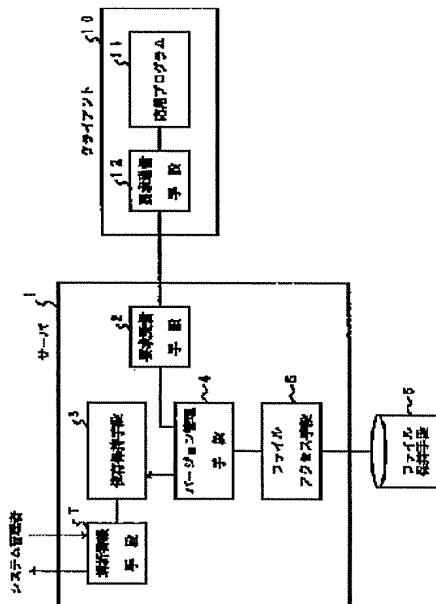
(54) **VIRUS PROTECTION SYSTEM**

(57) Abstract:

PURPOSE: To realize a virus protection system estimating the range of virus infection and restoring an infected program.

CONSTITUTION: A file change is performed without damaging an existing file by the version control with a version control means 4. Thus, the version for which a change is added due to a virus infection, etc., is deleted any time and the previous state can be restored. By using the dependence of files between clients controlled by a dependence holding means 3 and the history of the file access, a file having the possibility of the virus infection can be alarmed to a system controller without fail. The version infected with virus is deleted without fail since the version is restored based on the inspection result. Thus, virus infection can be completely protected.

COPYRIGHT: (C)1994,JPO&Japio



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-110718

(43)公開日 平成 6 年(1994) 4 月22日

(51)Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/00	3 1 0 N	7313-5B		
9/06	4 1 0 P	9367-5B		
12/00	5 3 1 R	8526-5B		

審査請求 未請求 請求項の数 3 (全 15 頁)

(21)出願番号 特願平4-261336

(22)出願日 平成 4 年(1992) 9 月30日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 関戸 一紀

神奈川県川崎市幸区小向東芝町 1 番地 株  
式会社東芝総合研究所内

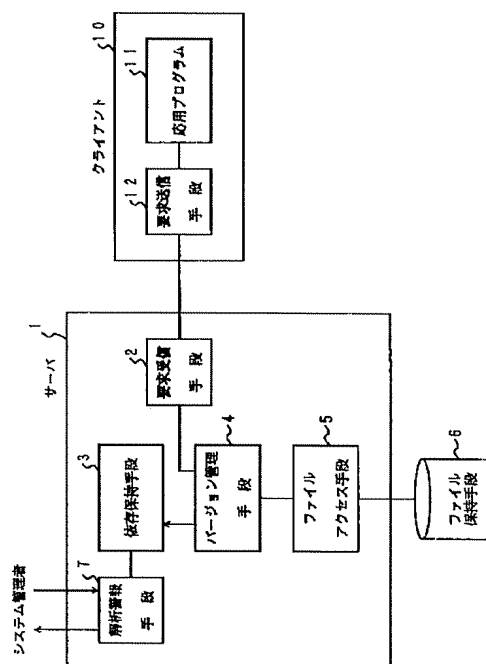
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 ウィルス防御方式

(57)【要約】

【目的】ウィルス感染の範囲を推定すると共に、感染したプログラムを元に戻すウィルス防御方式を実現する。

【構成】バージョン管理手段4によるバージョン管理によって、既存のファイルを損なうことなくファイル変更が行なわれる。よって、いつでもウィルス感染等によって変更が加えられたバージョンを削除し、前の状態に戻すことができる。また、依存保持手段3が管理しているクライアント間のファイルの依存関係およびファイルアクセスの履歴を用いることにより、ウィルス感染の可能性のあるファイルを必ずシステム管理者に警報できるとともに、その検査結果に基づいてバージョンをもとに戻しているの、ウィルスが感染したバージョンは必ず削除される。よって、ウィルス感染を完全に防御できる。



## 【特許請求の範囲】

【請求項1】 ファイルアクセスを要求する複数のクライアントと、これら各クライアントからのファイルアクセスに応じてファイルアクセスを行うサーバとから構成されるクライアント／サーバシステムにおいて、前記サーバは、前記各クライアントからのアクセス要求に従って、前記複数のクライアントのファイルアクセスの履歴およびクライアント間におけるファイルアクセスの依存関係を管理する第1の管理手段と、前記各クライアントからのファイル変更要求に応じて変更対象ファイルのバージョンを追加作成し、各ファイル毎にバージョンを管理する第2の管理手段と、前記第1の管理手段で管理されているファイルアクセスの履歴を解析してウィルス検査の必要性の有無を検出すると共に、前記クライアント間におけるファイルアクセスの依存関係に従って検査対象のファイルおよびクライアントを追跡する解析手段と、

所定のウィルス検査の結果にしたがって前記各ファイル毎に不要バージョンを削除する手段とを具備することを特徴とするクライアント／サーバシステムのウィルス防御方式。

【請求項2】 ファイルをアクセスを要求する応用プログラムと、そのアクセス要求をネットワーク経由でサーバに伝える要求送信手段から構成される複数のクライアントと、クライアントからのアクセス要求を受け取る要求受信手段と、この要求に従ってファイルアクセスを行うファイルアクセス手段と、ファイルを記憶しているファイル保持手段とから構成されるサーバとをネットワークで接続したクライアント／サーバシステムにおいて、前記サーバに、ファイル変更（挿入、削除、更新）が必要な場合にそのファイルの新しいバージョンを作成管理し、ファイル保持手段内のすでに存在するファイルを損なうことなくファイル変更を実現するとともに、指定された不必要なバージョンを削除するバージョン管理手段と、バージョン管理手段が行ったファイルアクセスの履歴から、ファイルを挿入・更新したクライアントとそのファイルを参照したクライアントとの間の依存関係と、各クライアント内でのファイルアクセスの順序関係によるファイル間の依存関係を作成保持する依存保持手段と、依存保持手段の情報を常に解析し、ウィルス検査の必要がないか調べ、必要な場合にはシステム管理者にすべて検査すべきクライアントとファイルを指定して警報するとともに、システム管理者がウィルス検査した結果を元に解析し、不必要となったバージョンをバージョン管理手段に伝え、さらに検査の必要なクライアントやファイルをシステム管理者に警報する解析警報手段を有し、システム管理者が解析警報手段の指示に従いクライアントやファイルのウィルス検査を行い、その結果を解析警

報手段に教えることを特徴とするウィルス防御方式。

【請求項3】 前記サーバは、さらにウィルス検査手段を有することを特徴とする請求項2記載のウィルス防御方式。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はクライアント／サーバシステムにおけるサーバのウィルス防御方式に関する。

【0002】

【従来の技術】近年、低価格で高性能なパソコン（PC）・ワークステーション（WS）の出現とネットワーク技術の進歩により、ファイルをクライアント／サーバ方式で共有するシステムが増えている。このシステムは、図12に示すように、独立して動作するのに必要なファイルしか持たない多数の小型計算機（クライアント）16-1～Nと、大量のファイルを保持する少数の（1つだけでもよい）大型計算機（サーバ）15をネットワーク17で接続した構成である。この方式により、これまで個々の計算機で重複して保持していたファイルを、1カ所に集中管理でき、クライアントが変わってもファイルをコピーして移動させる必要がない。さらに、クライアントには、大型のディスクを搭載する必要がなく、安価でコンパクトな計算機を使える。

【0003】また、最近、PCを中心にウィルスと呼ばれる計算機に障害を起こすプログラムによる被害が色々と報告されている。このプログラムは、予め設定された時期になると計算機内のファイルを消去したりOSの基本的なプログラムを破壊したりするもので、プログラムやOS等を経由して他のプログラムに伝染する機能を持っている。ウィルスに感染したプログラムを実行すると、まず、他のプログラムにもウィルスを感染させるため、ウィルスプログラムはその計算機からアクセス可能な全ファイル／ディレクトリを調べ、ウィルスの感染可能なプログラムやウィルス感染したプログラムを格納可能なディレクトリを見つける。また、感染可能なプログラムを順次読み出してウィルスプログラムを組み込んでから書き戻したり、ウィルス感染したプログラムを格納可能なディレクトリに書き込む。さらに、その計算機のプログラム実行を制御するOSにもウィルスプログラムを組み込み、OSが起動される毎に感染を試みる。なお、ウィルスチェッカと呼ばれるウィルスの存在を確認するプログラムも開発されている。

【0004】これまでは、フロッピー・ディスク等の媒体を介してウィルス感染する場合が主な経路で、各計算機の管理者が身元のはっきりしないプログラムを実行しないように注意すれば良く、その被害もウィルスプログラムを実行した計算機に限られた。しかし、クライアント／サーバ方式のシステムでは、ユーザのファイルはすべてサーバに格納されており、各クライアントからそれぞれにアクセスできる必要がある。よって、クライアント

からサーバ内の全ファイルが原則としてアクセス可能であり、1人の不注意なユーザによるウィルス感染したプログラムの実行が、サーバ内の全ファイルに感染し、さらに全クライアントに障害をもたらす危険性がある。よって、ウィルス防御手段がファイルサーバには必須である。しかし、従来のファイルサーバにはアクセス権による方法しかなく不十分であった。以下、従来のファイルサーバで採られているアクセス権方式とその問題点について簡単に説明する。

【0005】クライアント／サーバシステムにおける、ファイルサーバ20とクライアント25の構成を図13に示す。ファイルサーバ20は、クライアントからのファイル要求を受け取る要求受信手段21と、ファイルへのアクセス権を検査するアクセス権検査手段22と、ファイルやそのアクセス属性の読み出し書き込みを行うファイルアクセス手段23と、実際に多数のファイルを保持するファイル保持手段24から構成される。一方、クライアント25は、サーバ内にあるファイルを必要とする応用プログラム26と、現在クライアント25を使っているユーザの識別子を保持するユーザ識別子保持手段27と、応用プログラム26の指示に従ってサーバへ要求を送り出す要求送信手段28から構成される。なお、ユーザ識別子保持手段27内の識別子は、クライアント25がサーバ20へ論理的接続（ログインやマウント）を行った時に設定される。論理的接続ではパスワード検査等を行い、正規のユーザであることを確認している。

【0006】このシステムの動作について簡単に説明する。まず、クライアントの応用プログラム26が、ファイル名前とアクセス種別（読み出し／書き込み）を指定してアクセス要求を要求送信手段28に伝える。この要求を受け取った要求送信手段28は、ユーザ識別子保持手段27からユーザ識別子を読み出して、ファイル名、アクセス種別、ユーザ識別子からなるファイル要求を作成してサーバ20に送る。

【0007】サーバ20側では、この要求を要求受信手段21が受け取り、この要求内容に従って、ユーザ識別子のユーザが、ファイル名のファイルへ、アクセス種別のアクセスを行うものとして、ファイルアクセス手段23へアクセス要求を伝える。この要求を受け取ったファイルアクセス手段23は、まず指定されたファイルのアクセス属性をファイル保持手段24から取り出し、アクセス種別、ユーザ識別子とともにアクセス権検査手段22に送る。アクセス権検査手段22では、格納されていたアクセス属性を解析し、このファイルはユーザ識別子のユーザに対してアクセス種別のアクセスが許されているか検査する。この結果がファイルアクセス手段23に伝えられ、許可される場合にのみアクセス種別のアクセスがファイル保持手段24内のファイル名のファイルに対して行われる。

【0008】ここで、応用プログラム26がウィルスに

感染している場合を考える。応用プログラムを実行するとウィルスプログラムが動き出す。ウィルスプログラムはサーバ内のファイルへウィルスを感染させようと試み、ファイル名とアクセス種別を指定してアクセスを要求送信手段28に伝える。要求送信手段28は、ユーザ識別子保持手段27から現在クライアントを使っているユーザのユーザ識別子を読み出して、アクセス要求を作成してサーバ20に送る。

【0009】サーバ20側では、この要求を要求受信手段21が受け取り、ファイルアクセス手段23へ伝える。ファイルアクセス手段23は、まず指定されたファイルのアクセス属性をファイル保持手段24から取り出して、アクセス種別やユーザ識別子とともにアクセス検査手段22に送る。アクセス検査手段22では、格納されていたアクセス属性を解析し、このファイルがユーザ識別子のユーザにアクセス種別のアクセスが許可されているかを検査する。許可されている場合にのみ、ファイルアクセス手段23はファイル保持手段24内のファイルに対してアクセスを行う。よって、各ユーザがアクセス（特に変更）できるファイル／ディレクトリの属性を非常に限られたものに設定することにより、サーバ内の大部分のファイルにはアクセス検査手段22からアクセス許可が出ず、ウィルスプログラムが感染できるプログラムは非常に制限される。

【0010】以上のように、サーバ側ではユーザ識別子とアクセス属性を用いてファイルへのアクセスを制限しており、結果としてウィルスプログラムを実行したユーザが変更を許されているプログラムにしか影響が及ばない。よって、ファイルへのアクセス権方式を用いて、ウィルスプログラムを実行したユーザの範囲内にウィルス汚染の影響を抑えられる。しかし、本方式には次のような問題点がある。

【0011】第1に、ウィルスプログラムを実行したユーザが管理している他のプログラムやディレクトリにはアクセス許可が常に出ているので、それにウィルスが感染するのを防げない。

【0012】第2に、一般にはユーザ毎に別々にプログラムやディレクトリが存在するのではなく、共通のプログラムやディレクトリ、即ちウィルス感染の可能なプログラムやウィルスプログラムを格納可能なディレクトリが存在する。よって、あるユーザがウィルスプログラムを実行すると、共通のプログラムに感染していたり、ディレクトリにウィルス感染プログラムを格納したりする。よって、それ以降にそのプログラムを実行したユーザにもウィルス汚染が及んでしまう。

【0013】第3に、ウィルスプログラムを実行してクライアントのOSに感染した場合、そのクライアントをそのまま他のユーザが使うと、OS内のウィルスプログラムが作動してそのユーザにアクセスの許されているプログラム／ディレクトリにまで感染してしまう。

【0014】第4に、このようなサーバには先のアクセス権によってアクセスが禁止されない特権ユーザが存在しており、クライアント／サーバシステムを管理・運用している。よって、特権ユーザとしてウィルスプログラムをクライアントかサーバで実行すると、サーバ内の全プログラムに感染させることが可能となる。さらに、これらの感染原因は複合して作用し、一層感染の範囲を広げてしまう。

【0015】また、サーバ内の全ファイルと全クライアントに感染する可能性があるので、ウィルスチェッカがウィルスが存在しないことや感染が及んでいる範囲を確認するには、全ファイルと全クライアントを検査する必要があり、ファイル数やクライアント数に比例した膨大な作業を必要とする。

【0016】

【発明が解決しようとする課題】以上のように、従来のアクセス権方式ではサーバ内にウィルスが感染するのを完全に防げないばかりでなく、サーバ内にウィルスが存在しないことやウィルス感染の範囲を確認するのに膨大な作業が必要であった。本発明は、ウィルス感染の範囲を推定するとともに感染したプログラムを元にもどすウィルス防御方式を提供することを目的とする。

【0017】

【課題を解決するための手段および作用】本発明のウィルス防御方式は、ファイルアクセスを要求する複数のクライアントと、これら各クライアントからのファイルアクセスに応じてファイルアクセスを行うサーバとから構成されるクライアント／サーバシステムにおいて、前記サーバに、前記各クライアントからのアクセス要求に従って、前記複数のクライアントのファイルアクセスの履歴およびクライアント間におけるファイルアクセスの依存関係を管理する第1の管理手段と、前記各クライアントからのファイル変更要求に応じて変更対象ファイルのバージョンを追加作成し、各ファイル毎にバージョンを管理する第2の管理手段と、前記第1の管理手段で管理されているファイルアクセスの履歴を解析してウィルス検査の必要性の有無を検出すると共に、前記クライアント間におけるファイルアクセスの依存関係に従って検査対象のファイルおよびクライアントを追跡する解析手段と、所定のウィルス検査の結果にしたがって前記各ファイル毎に不要バージョンを削除する手段とを具備することを特徴とする。

【0018】このシステムにおいては、第2の管理手段によるバージョン管理によって、既存のファイルを損なうことなくファイル変更が行なわれる。よって、いつでもウィルス感染等によって変更が加えられたバージョンを削除し、前の状態に戻すことができる。また、第1の管理手段が管理しているクライアント間のファイルの依存関係およびファイルアクセスの履歴を用いることにより、ウィルス感染の可能性のあるファイルを必ずシス

テム管理者に警報できるとともに、その検査結果に基づいてバージョンをもとに戻しているため、ウィルスが感染したバージョンは必ず削除される。よって、ウィルス感染を完全に防御できる。

【0019】さらに、解析手段が常に情報を解析してウィルス存在の可能性をチェックしているため、ウィルス感染しても広範囲に広がる前に発見できる。また、クライアント間の依存関係の情報を使ってウィルス感染の可能性のあるクライアントやファイルを限定できる。よって、サーバ内におけるウィルスの存在や存在するときの感染範囲を確認するのに少数のファイルやクライアントを検査するだけでよい。

【0020】

【実施例】以下に本発明の実施例を説明する。図1は本発明のウィルス防御方式を適用して構成した、サーバ1とクライアント10からなるクライアント／サーバシステムの構成図である。

【0021】サーバ1は、クライアントからのアクセス要求を受け取る要求受信手段2と、クライアントからのアクセス要求に従ってクライアント間やファイル間の依存関係を作成保持する依存保持手段3と、ファイルへの変更（挿入、削除、更新）をファイルに直接反映せずバージョンを作って管理するバージョン管理手段4と、ファイルの読み書きを行うファイルアクセス手段5と、実際に多数のファイルを保持するファイル保持手段6と、依存保持手段3が保持しているクライアント間やファイル間の依存関係を解析しウィルス検査の必要性をシステム管理者に警報する解析警報手段7から構成される。一方、クライアント10は、サーバ内にあるファイルを必要とする応用プログラム11と、応用プログラム11の指示に従いサーバ2へアクセス要求を送り出す要求送信手段12から構成される。このシステムの動作について以下で簡単に示す。

(1) クライアントの応用プログラム11がファイル名とアクセス種別（読み出し、書き込み）からなるアクセス要求を要求送信手段12に伝える。

(2) この要求を受け取った要求送信手段12は、クライアント名を含むファイル要求を作成してサーバ1に送る。

(3) サーバ1では、この要求を要求受信手段2が受け取り、ファイル名、アクセス種別およびクライアント名をバージョン管理手段4に伝える。

【0022】(4) バージョン管理手段4は、指定されたファイル名のファイルがすでに管理されているか調べる。もし、管理されていないならば、ファイル保持手段6に存在するかをファイルアクセス手段5に問い合わせた後、管理対象ファイルとして登録する。

【0023】(5) 次に、バージョン管理手段4は、登録された情報に従って、指定されたファイルにアクセス種別のアクセスが可能か調べ、そのアクセス形態（読み

出し、挿入、更新、削除)を決定する。なお、アクセス可能な場合にのみアクセス処理が継続する。

【0024】(6)さらに、バージョン管理手段4は、確定したアクセス形態、ファイル名、クライアント名を依存保持手段3に伝えとともに、既存ファイルの内容を損なわないよう、バージョンを作るファイル要求に変えてファイルアクセス手段5に伝える。

【0025】(7)ファイルアクセス手段5は、要求されたファイルアクセスをファイル保持手段6内のファイルに対して行い、その結果がバージョン管理手段4、要求受信手段2、要求送信手段12を経由して応用プログラム11に伝えられる。

【0026】(8)一方、確定したアクセス形態、ファイル名、クライアント名を受け取った依存保持手段3は、その情報を元にクライアント間やファイル間の依存関係を作成保持し、その情報を解析警報手段7に提供する。

【0027】(9)解析警報手段7は依存保持手段3の情報を常に解析し、ウィルス検査の必要がある場合にはシステム管理者に検査すべきクライアントやプログラムとともに警報する。また、システム管理者からのウィルス検査の結果に基づき、ウィルス感染の危険性が無くもはや必要なくなったバージョンやウィルスに感染していたファイルのバージョンをバージョン管理手段4に伝える。

【0028】(10)バージョン管理手段4は、もはや必要が無くなったバージョンの削減と、ウィルスに感染したファイルの古いバージョンへの復元を行う。また、登録されているそのファイルに関する情報を更新する。

【0029】ここで、応用プログラム11がウィルスに感染しており、ファイルA、B、Cへこのウィルスが伝染する場合を考える。応用プログラムを実行するとウィルスプログラムが動き出す。ウィルスプログラムはサーバ内のファイルA、B、Cへ感染を試み、図2に示す順序でファイル名とアクセス種別を指定してアクセス要求を要求送信手段12に伝える。要求送信手段はこれらアクセス要求からサーバ1へのファイル要求を作成して送

【0030】サーバ1側では、これらの要求を要求受信手段2が受け取り、そのファイル名、アクセス種別、クライアント名をバージョン管理手段4に伝える。バージョン管理手段4は、これらのファイルをバージョン管理されるものとして登録し、そのアクセス形態(参照、挿入、更新、削除)を決定する(図3参照)。さらに、アクセス形態が更新の場合には、その更新要求を、既存のファイル内容を失わないように新しいバージョンファイルを作る、削除の場合には何もしない、といったアクセス要求に変えてファイルアクセス手段5に伝える。ファイルアクセス手段5は、これらのアクセスをファイル保持手段6内のファイルに対して行い、その結果をバージ

ョン管理手段4、要求受信手段2、要求送信手段12を経由して応用プログラム11に伝える。よって、クライアントからのアクセス要求の処理後には、図4に示すように、ファイルA、B、Cはそれぞれバージョン $a_1$ 、 $b_1$ 、 $c_1$ からウィルスに感染したバージョン $a_2$ 、 $b_2$ 、 $c_2$ に変わる。しかし、バージョン $a_1$ 、 $b_1$ 、 $c_1$ のファイルも変更が加えられずに残っているので、あとでバージョン $a_1$ 、 $b_1$ 、 $c_1$ に戻すことで復元が可能である。

【0031】また、バージョン管理手段4は、決定したアクセス形態、ファイル名、クライアント名を依存保持手段3に伝える。依存保持手段3は、これまでに保持していたクライアント間とファイル間の依存関係にこれらの情報を追加する。よって、クライアント10におけるファイル間の依存関係の情報として図3が追加され、クライアント10についてのファイル依存関係の情報は図5のようになる。

【0032】解析警報手段7はこの依存関係の情報を常に解析し、ウィルス存在の可能性を示すアクセスパターンがないか調べる。もし、該当するアクセスパターンが発見されウィルス検査の必要がある場合には、システム管理者に検査すべきクライアントやファイルを指定して警報する。よって、この例の場合、図5に示すように3つのファイル更新が短時間に連続して行われていることが発見され、ウィルス検査すべきクライアントとしてクライアント10、検査すべきファイルとしてP(Pは応用プログラム11が格納されているファイル)、A、B、Cがシステム管理者へ警報される。システム管理者は、ウィルスチェッカなどの手段で、ウィルスの存在を調べ、ファイルP、A、B、C内のウィルスを発見する。この結果をシステム管理者は解析警報手段7に入力する。解析警報手段7は、ウィルスに感染していた場合にはそのファイルの新しいバージョンを、感染していなかった場合には古いバージョンを不要なバージョンとしてバージョン管理手段4に伝え、不必要なバージョンを削除する。よって、この例ではファイルP、A、B、Cの新しいバージョン即ち、ファイルPの $p_1$ 、ファイルA、B、Cの $a_2$ 、 $b_2$ 、 $c_2$ がそれぞれ削除され、ウィルスが感染する前のファイルバージョンに復元される。

【0033】なお、これらのウィルスに感染したバージョン $p_1$ 、 $a_2$ 、 $b_2$ 、 $c_2$ のどれかを他のクライアントが参照していた場合、依存保持手段3には、図6のようなクライアント間の依存関係が保持されている。よって、解析警報手段7は、ウィルス感染したバージョン $p_1$ を参照しているクライアントBとその参照の後で更新されたファイルDを検査するようにシステム管理者にさらに警報する。このシステム管理者とのウィルス検査のやりとりはウィルス感染の可能性のあるバージョンがなくなるまで続けられる。

【0034】次に、従来のアクセス権方式で問題となった場合を考える。本方式では、バージョン管理手段4により既存のファイルを古いバージョンとして残し、システム管理者によりウィルスに感染していないことが確認された場合だけそれを削除している。ウィルスに感染したことが判った時はいつでもウィルス感染の影響を取り除ける。よって、第1の場合でもウィルス感染を防げる。また、依存保持手段3ではファイル間の依存関係とクライアント間依存関係を保持しており、解析警報手段7は第2の場合でも第3の場合でもその感染の可能性のあるファイルを追跡でき、システム管理者の検査によりウィルス感染の影響を取り除ける。さらに、本方式はアクセス権を全く使っていないので、第4の場合でも防衛できる。最後に、本方式では、依存保持手段3の依存情報を元に感染の可能性のあるクライアントやファイルを求めるとともに、解析警報手段7が常に依存情報を解析してウィルスの感染が大きく広がる前にシステム管理者に警報するので、ウィルスチェックなどでウィルスが存在しないことや感染が及んでいる範囲を確認するにも少数のファイルやクライアントを検査するだけでよい。最後に、本実施例を構成する各要素の具体例について説明する。

【0035】要求送信手段12と要求受信手段2は、クライアント／サーバシステムに必須の構成要素である。UNIXのクライアント／サーバ型ファイルシステムNFSを例にとると、要求送信手段12はNFSクライアントとその通信プログラム(XDR, RPC, UDP, IP)に相当し、要求受信手段2はNFSサーバとその通信プログラムに相当し、その仕様はTCP/IPの標準化組織(IAB: Internet Activities Board)からRFCとして公開されている。よって、当業者にとっては自明であるのでこれ以上の説明はここでは省略する。

【0036】ファイル保持手段6は磁気ディスク、又は光ディスク、又は半導体メモリ等からなる不揮発性の記憶媒体で、多数のファイルを保持し、その読み出し／書き込みが可能なものである。一般には、磁気ディスク装置が用いられる。

【0037】ファイルアクセス手段5は、ファイル名(またはその識別子)からファイル保持手段内におけるそのファイルの格納位置を求め、アクセス要求に従ってファイルの読み出し／書き込みを行うものである。UNIXを例にとると、ファイルシステムがこれに相当し、その詳細は文献「UNIXカーネルの設計：共立出版、Maurice J Bach著」に述べられている。

【0038】バージョン管理手段4は、ファイルへの変更(挿入、削除、更新)をファイル保持手段6内のファイルを損なうことなく実現するため、バージョン(版)を作成してその情報を管理する。バージョン情報を管理するバージョン情報テーブルは、図7に示すように、フ

ファイル名フィールドと、複数のバージョンフィールドとから構成されており、ユーザから見えるファイルのファイル名と、その各バージョンの実際のファイル名の対応関係を保持している。バージョンフィールドの一番左のファイル名がもっとも古いバージョンを意味し、もっとも右が最新のバージョンを意味する。また、(deleted)は削除されたことを意味する。よって、図7においてDEFファイルは削除されていることを表わす。このテーブルを使ったバージョン管理手段4の概略フローを図8に示す。

【0039】すなわち、バージョン管理手段4は、まず、指定されたファイル名のファイルがすでに管理されているか調べる(ステップS1)。もし、管理されていなければ、ファイル保持手段6に存在するかをファイルアクセス手段5に問い合わせた後、管理対象ファイルとして登録する(ステップS2)。次に、バージョン管理手段4は、登録された情報に従って、指定されたファイルにアクセス種別のアクセスが可能か調べ、そのアクセス形態(読み出し、挿入、更新、削除)を決定する(ステップS3, S4)。なお、アクセス不可能な場合には、エラーとなり処理が中断される。

【0040】さらに、バージョン管理手段4は、確定したアクセス形態、ファイル名、クライアント名を依存保持手段3に伝えたとともに、既存ファイルの内容を損なわないよう、バージョンを作るファイル要求に変えてファイルアクセス手段5に伝える(ステップS5)。例えば、アクセス形態が参照の場合には、図7のテーブルのフィールドがサーチされ、最も右側の最新のバージョンが見つけれ、それがファイルアクセス手段5によって読み出される(ステップS6)。アクセス形態が挿入の場合には、新しいバージョン名のファイルが作成されてファイル保持手段6に登録されると共に、そのバージョン名がバージョンフィールドの先頭に書き込まれる(ステップS7)。アクセス形態が削除の場合には、フィールドの最右端に(deleted)が書き込まれる(ステップS8)。アクセス形態が更新の場合には、新しいバージョン名のファイルが作成され、それがファイル保持手段6に登録されると共に、新しいバージョン名がフィールドの最右端に書き込まれる(ステップS9)。なお、ウィルス検査の結果、もはや不要になったバージョンはテーブルからその名前が削除されるときに、ファイル保持手段6からも実際に削除される。

【0041】依存保持手段3は、バージョン管理手段5からのアクセス形態、ファイル名、クライアント名に基づいて、クライアント間とファイル間の依存関係を作成保持する。依存関係を保持する依存関係テーブルは、図9に示すように、各クライアント毎に存在し、ファイル名、アクセス形態、ポインタフィールドから構成されている。各テーブルはそのクライアントで実行されたファイルアクセスの内、ウィルス感染の範囲を限定するのに

必要なものが格納されている。図10に依存保持手段3の依存関係テーブルを作成するフローを示す。なお、ウィルス検査の結果、もはや不要になった情報は削除される。

【0042】すなわち、まず、アクセス形態が参照か否かが判断され（ステップS11）、参照以外の場合には、クライアント名で指定されたテーブルにファイル名と対応するアクセス形態が追加される（ステップS12）。一方、アクセス形態が参照の場合には、指定されたファイルの更新、挿入がテーブルにすでにあるか否かが判断され（ステップS13）、存在する場合には、ク

ライアント名で指定されたテーブルに、ファイル名とアクセス形態が追加されると共に、ステップS13で見つけた更新、挿入のエントリ間がポインタで結合される（ステップS14）。解析警報手段7は次の4つを処理する。

（1）依存保持手段3の情報を常に解析し、次に示すようなウィルス検査が必要なアクセスパターンがあるか調べる。

パターンA； ウィルス動作の可能性を示すアクセスパターン

a. サーバ全体で予め定められた頻度以上のファイル変更が行われる。

b. 1つのクライアントで予め定められた頻度以上のファイル変更が行われる。

c. 予め定められた名前のファイルが変更される。

パターンB； ウィルスの存在が判明したとき、その影響が広範囲に及ぶ可能性があるアクセスパターン

a. 1つのクライアントで累積のファイル変更が予め定めた回数を越える。

b. 1つの変更されたファイルで累積の参照が予め定めた回数を越える。

（2）ウィルス検査が必要な場合、ウィルス検査するクライアントとファイルを次のようにして求めて、システム管理者に警報する。

【0043】パターンAの場合は、アクセスパターン内で変更されているファイル、アクセスパターンの中で参照されたファイル、アクセスパターンが発見されたクライアントを検査対象にする。パターンBの場合は、アクセスパターンが発見されたクライアント、又はアクセスパターンに含まれるファイルを検査対象にする。

（3）システム管理者が行ったウィルス検査の結果に基づいて、次のようにしてもはや不要になったバージョンを求めてバージョン管理手段に伝える。

・ウィルスに感染したくないことがわかった場合、そのバージョンより前のものを不要なバージョンとする。

・ウィルスに感染していることがわかった場合、そのバージョンより後のものを不要なバージョンとする。

【0044】（4）システム管理者が行ったウィルス検査の結果に基づいて、次のようにしてさらにウィルス検

査が必要なファイルやクライアントを求めシステム管理者に警報する。

・ウィルスが発見されたファイルを参照したクライアント。

・ウィルスが発見されたクライアントで変更されたファイル。

【0045】なお、この実施例では、ウィルスに感染した可能性を検出する機能と感染していた場合に元に戻す機能が提供されているだけである。よって、感染したか否かの判断は、システム管理者等がウィルスチェッカ等の別の手段を使うことを想定していた。これではシステム管理者の仕事が大変になってしまうので、次に、ウィルスに感染したか否かを判断する感染判断機能について述べる。

【0046】ウィルス感染をサーバで検出するには、まず、クライアント又はサーバにウィルスが何時でも感染させられるような被感染ファイルを用意し、クライアント側にある被感染ファイルの内容をウィルスが感染する前にサーバ側にコピーを取って置く。そして、サーバ側にある被感染ファイルに対してはそのファイルが更新アクセスが起きないか、クライアント側にある被感染ファイルに対しては定期的にサーバ内にあるコピーと比較して変更された形跡がないかを調査する。もし、更新アクセスや更新された形跡があれば、その更新アクセスに出したクライアントやその更新された形跡のあるクライアントはウィルスに感染している事を意味し、ウィルス感染を検出できる。以下、図11を参照して具体的に説明する。図11はウィルス検出のためのシステム構成である。

【0047】クライアントには、誰でもアクセスできる被感染ファイルBが存在する。サーバには、誰でもアクセス出来る被感染ファイルAと、その被感染ファイルAに対する更新を監視するファイル更新監視手段と、クライアントにある被感染ファイルBのコピーファイルと、定期的にクライアント内の被感染ファイルBの内容とサーバ内の被感染ファイルBのコピーファイルの内容が一致するか比較監視するファイル比較監視手段が存在する。ここで、クライアントでウィルスプログラムが起動された場合を考える。

【0048】（1）まず、ウィルスプログラムは、自分自身を感染させられるファイルを探す。ここで、ウィルスプログラムは、サーバ内の誰でもアクセス出来る被感染ファイルAやクライアント内の誰でもアクセス出来る被感染ファイルBを発見する。

（2）次に、ウィルスプログラムは、発見したファイルに対してウィルス感染を試み、ファイルの更新を行なう。よって、被感染ファイルA、Bは更新され、ウィルスが感染してしまう。ここで、サーバ内にある被感染ファイルAに対する更新は、サーバ内のファイル更新監視手段により検出され、ファイル更新を要求したクライ



ントにウィルスプログラムが存在する事が判る。

【0049】(3)サーバ内のファイル比較監視手段は、定期的にクライアント内の被感染ファイルとサーバ内のそのコピーを比較し、一致しているか検査する。ここで、クライアント内の被感染ファイルBは、ウィルスプログラムによりステップ2で更新されているので、一致しない。よって、被感染ファイルBがあるクライアントにウィルスプログラムが存在することが判る。

【0050】以上のように、クライアントにウィルスプログラムが存在すれば、サーバ内の被感染ファイルAかクライアント内の被感染ファイルBのどちらかが変更され、ファイル更新監視手段かファイル比較監視手段によりウィルスの存在が検出される。

【0051】なお、被感染ファイルA、Bは、テキストファイルに限定するものではなく、コマンドファイル、ライブラリファイル、実行ファイル、OS自身を格納するファイルなど計算機に格納されるあらゆる種類のファイルを用いることができる。以上のように、この実施例においては、バージョン管理手段4によるバージョン管理によって、既存のファイルを損なうことなくファイル変更が行なわれる。よって、いつでもウィルス感染等によって変更が加えられたバージョンを削除し、前の状態に戻すことができる。また、依存保持手段3が管理しているクライアント間のファイルの依存関係およびファイルアクセスの履歴を用いることにより、ウィルス感染の可能性のあるファイルを必ずシステム管理者に警報できるとともに、その検査結果に基づいてバージョンをもとに戻しているのので、ウィルスが感染したバージョンは必ず削除される。よって、ウィルス感染を完全に防御できる。

【0052】さらに、解析警報手段7が常に情報を解析してウィルス存在の可能性をチェックしているので、ウィルス感染しても広範囲に広がる前に発見できる。また、クライアント間の依存関係の情報を使ってウィルス感染の可能性のあるクライアントやファイルを限定できる。よって、サーバ内におけるウィルスの存在や存在するときの感染範囲を確認するのに少数のファイルやクライアントを検査するだけでよい。

【0053】

【発明の効果】以上のように、本ウィルス防御機能では

サーバ内にウィルスが感染するのを完全に防げるばかりでなく、サーバ内にウィルスが存在しないことやウィルス感染の範囲を確認するのも少数のファイルを検査するだけでよい。よって非常に効率の良いウィルス対策を実現できる。

【図面の簡単な説明】

【図1】この発明の一実施例に係わるウィルス防御方式を実現するためのクライアント／サーバシステムの構成を示すブロック図。

【図2】同実施例のシステムにおけるクライアントからのアクセス要求の一例を示す図。

【図3】同実施例のシステムにおいてサーバで確認されるアクセス形態の一例を示す図。

【図4】同実施例のシステムにおいてサーバで管理されるバージョン関係の一例を示す図。

【図5】同実施例のシステムにおいてサーバで管理されるクライアントのファイル依存関係を示す図。

【図6】同実施例のシステムにおいてサーバで管理されるクライアント間の依存関係を示す図。

【図7】同実施例のシステムにおいてサーバで管理されるバージョン管理テーブルの一例を示す図。

【図8】同実施例のシステムにおいてサーバ内で実行されるバージョン管理動作の一例を示すフローチャート。

【図9】同実施例のシステムにおいてサーバで管理される依存関係テーブルの一例を示す図。

【図10】同実施例のシステムにおいてサーバ内で実行される依存関係テーブル作成動作の一例を示すフローチャート。

【図11】同実施例のシステムで採用されるウイルス検査方式の一例を説明するための図。

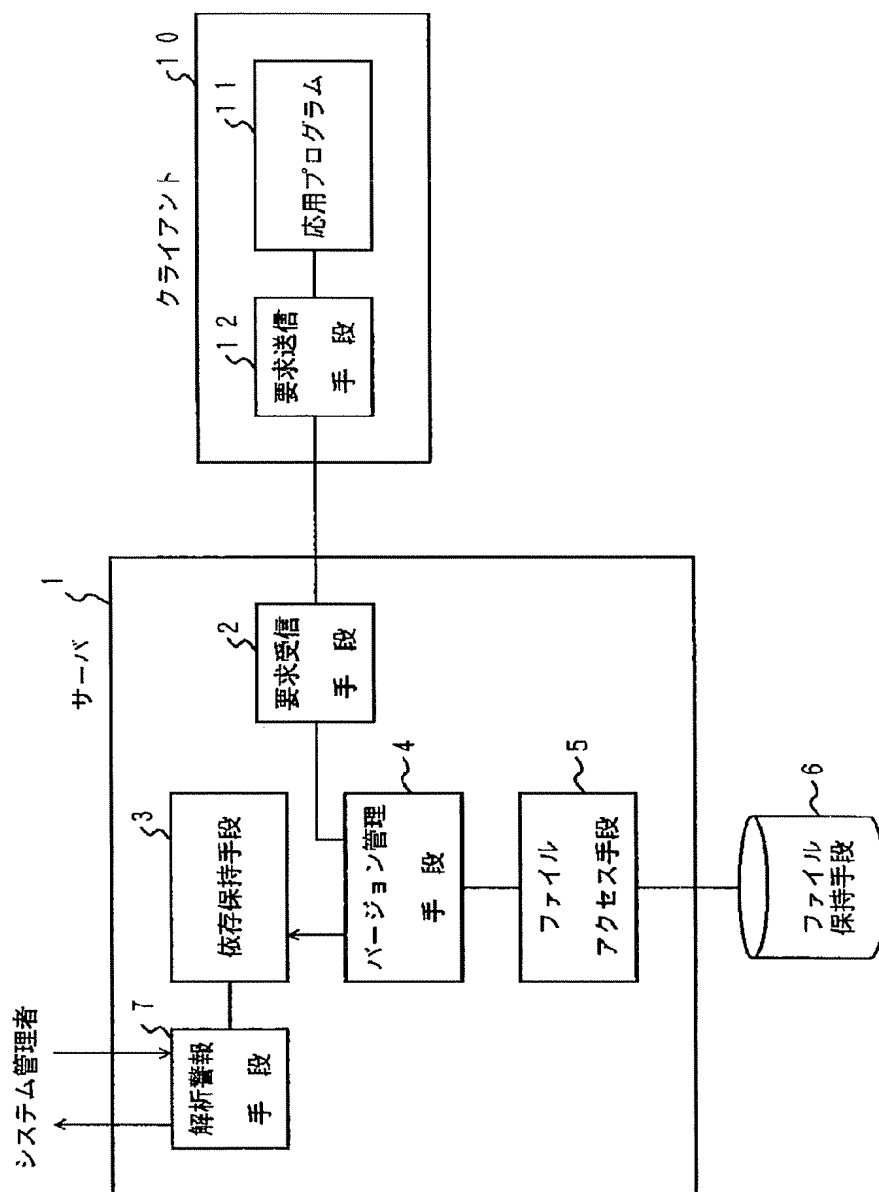
【図12】従来のクライアント／サーバシステムを概念的に示すブロック図。

【図13】従来のクライアント／サーバシステムにおけるファイルアクセス動作を説明するための図。

【符号の説明】

1…サーバ、2…要求受信手段、3…依存関係保持手段、4…バージョン管理手段、5…ファイルアクセス手段、6…ファイル保持手段、7…解析警報手段、10…クライアント。

【図1】



【図2】

ファイル名	アクセス種別
ファイルA	読み出し
ファイルA	書き込み
ファイルB	読み出し
ファイルB	書き込み
ファイルC	読み出し
ファイルC	書き込み

クライアントのアクセス要求

【図3】

ファイル名	アクセス形態
ファイルA	参照
ファイルA	更新
ファイルB	参照
ファイルB	更新
ファイルC	参照
ファイルC	更新

アクセス形態

【図4】

ファイル名	バージョン
P	p <sub>1</sub>
A	a <sub>1</sub> → a <sub>2</sub>
B	b <sub>1</sub> → b <sub>2</sub>
C	c <sub>1</sub> → c <sub>2</sub>

バージョン関係

【図5】

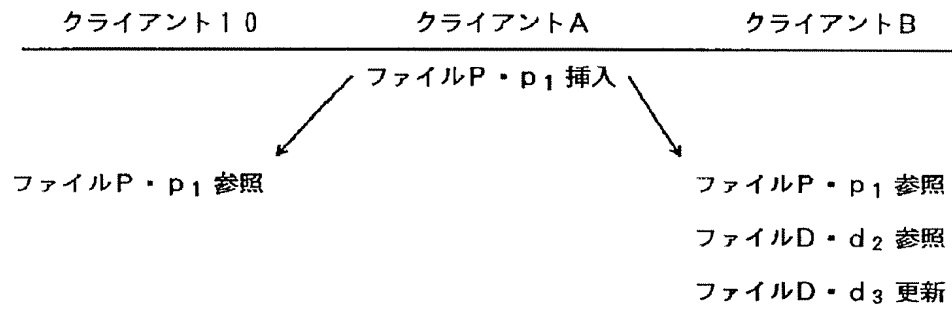
ファイル名	バージョン	アクセス形態
P	p <sub>1</sub>	参照
A	a <sub>1</sub>	参照
A	a <sub>2</sub>	更新
B	b <sub>1</sub>	参照
B	b <sub>2</sub>	更新
C	c <sub>1</sub>	参照
C	c <sub>2</sub>	更新

クライアント10のファイル依存関係

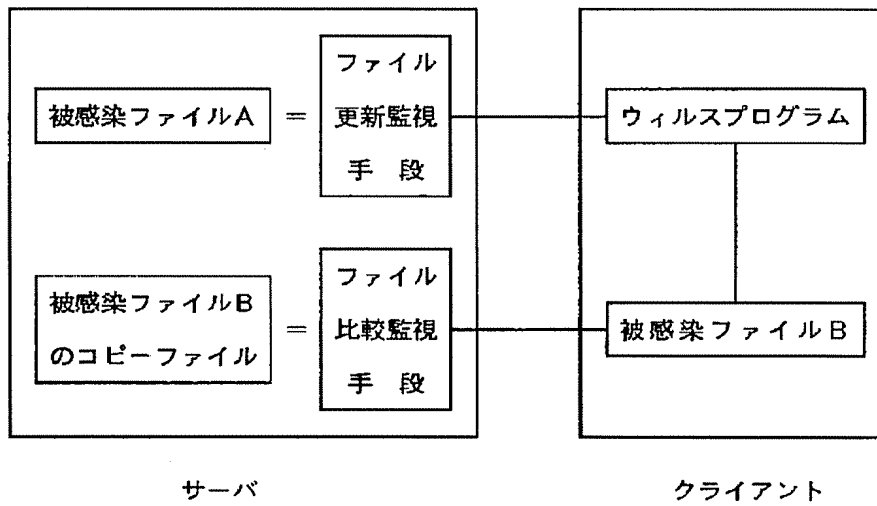
【図7】

ファイル名	バージョン1	バージョン2	バージョン3	バージョン4	バージョン5	.....
ABC	ABC	ABC. 1	ABC. 2	ABC. 3		
DEF	DEF	DEF. 1	(deleted)			
GHI	GHI. 3	GHI. 4				
KLM	(deleted)					

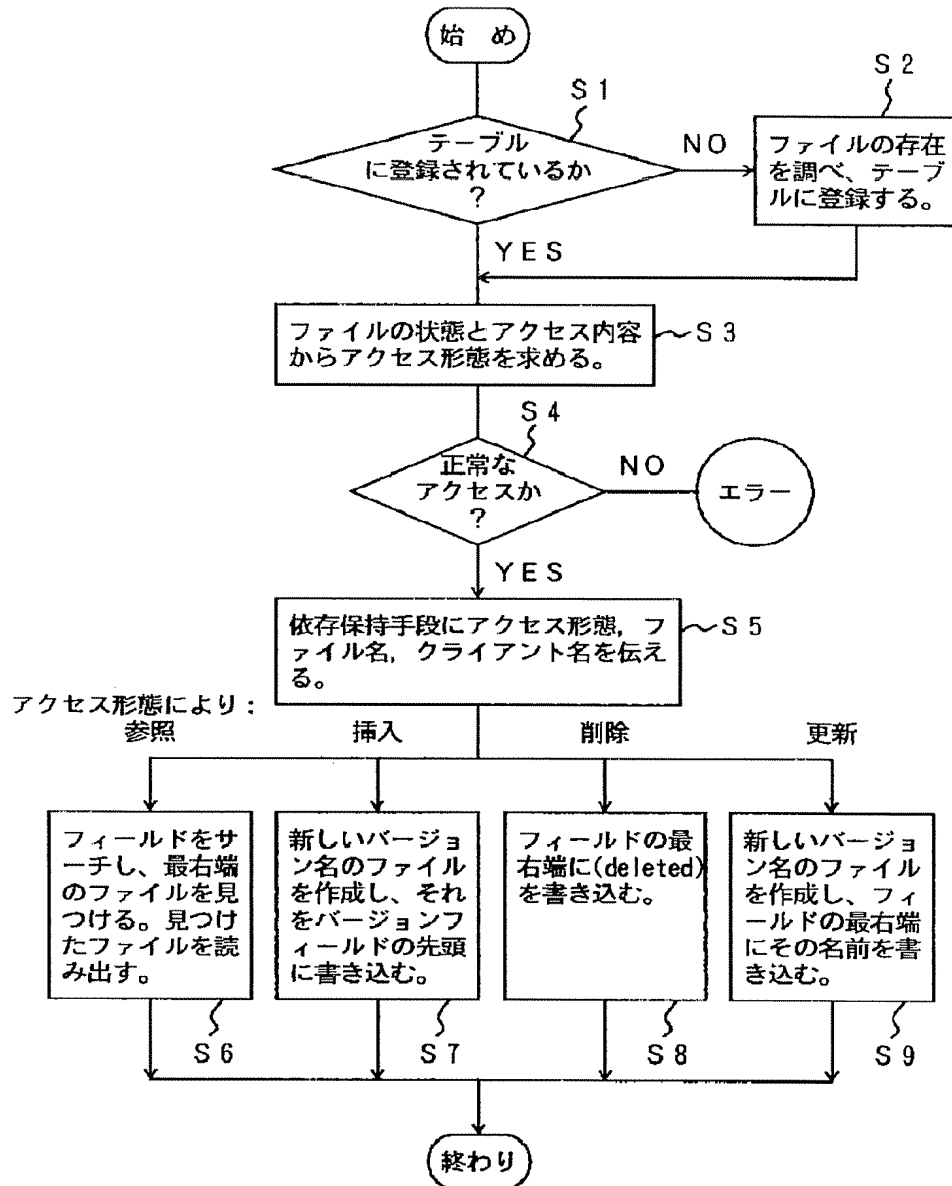
【図6】



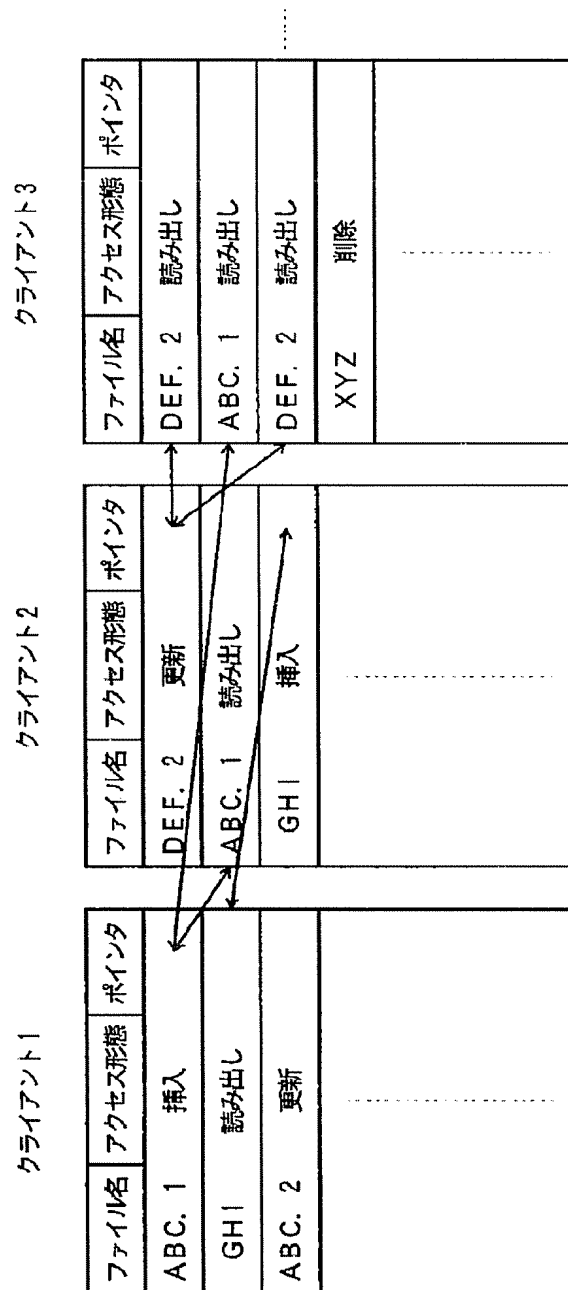
【図11】



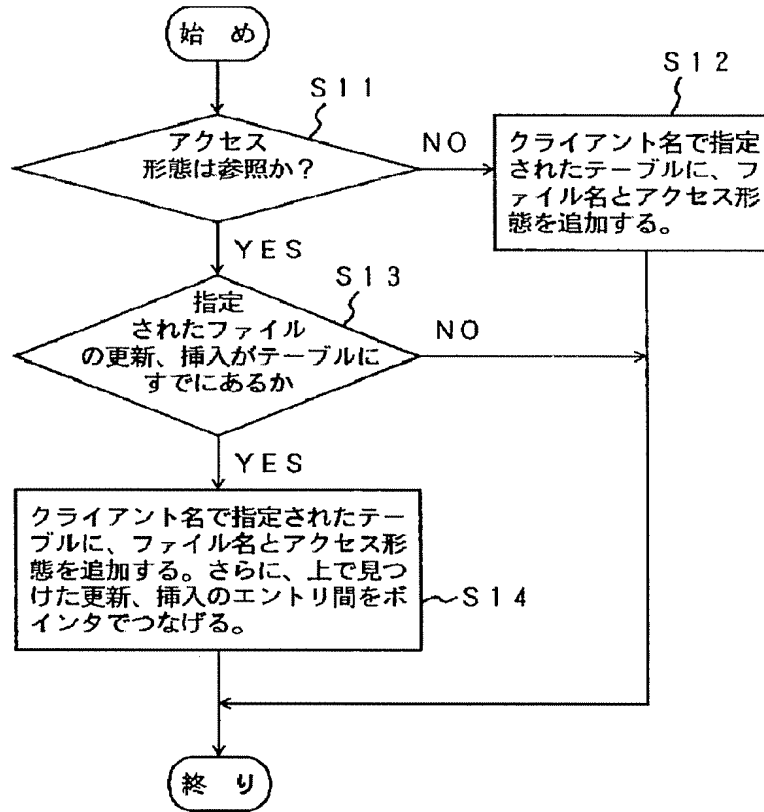
【図8】



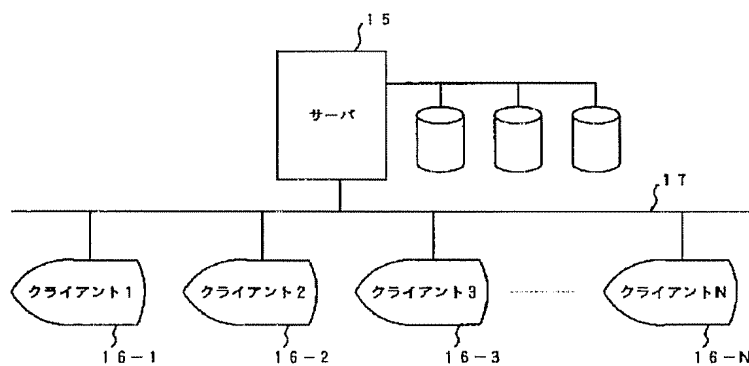
【図9】



【図10】



【図12】



【図13】

